

ESTUDIO DE LAS PRESTACIONES DE UN ALGORITMO LMS MODIFICADO: APLICACIÓN EN INGENIERÍA BIOMÉDICA.

SORIA OLIVAS, E.
FRANCÉS VÍLLORA, J.V.
SERRANO LÓPEZ, A.J.
MARTINEZ SOBER, M.
CALPE MARAVILLA, J.
GUERRERO MARTÍNEZ, J.

G.P.D.S.
GRUPO DE PROCESADO DIGITAL DE SEÑALES
Departamento de Informática y Electrónica.
Facultad de Físicas
C/ Dr Moliner, 50
46100 Burjassot, VALENCIA. ESPAÑA
e-mail: emilio.soria@uv.es

RESUMEN:

En este trabajo se plantea el estudio de un nuevo algoritmo adaptativo que presenta mejor comportamiento que el método de mínimos cuadrados clásico, LMS (Least Mean Square). Este algoritmo se basa en el aprovechamiento del tiempo muerto intermuestro para intercalar otro algoritmo de búsqueda. El algoritmo escogido es el de Fletcher-Reeves por su alta velocidad de convergencia. El procedimiento planteado ha demostrado su eficacia en un problema teórico como es la modelización de un sistema y en dos aplicaciones de ingeniería biomédica, la eliminación del ruido de red y la obtención del electrocardiograma (ECG) fetal.

PALABRAS CLAVE:

Filtrado adaptativo, LMS, Optimización.

ABSTRACT:

This work presents a new adaptive algorithm with improved performance over the classical Least Mean Square (LMS) method. This algorithm takes advantage of the lag time between samples to implement another search algorithm. The Fletcher-Reeves algorithm was chosen because of its high convergence rate. The proposed method has demonstrated its efficacy in a system modeling theoretical problem, and in two biomedical engineering applications: network noise filtering and fetal ECG extraction.

KEYWORDS:

Adaptive filtering, LMS, Optimization.

INTRODUCCIÓN.

Un sistema adaptativo intenta modelizar la diferencia que hay entre la salida del sistema y una señal que se considera como señal deseada, figura 1:

El algoritmo adaptativo LMS modifica los coeficientes del filtro para minimizar una función del error cometido que, normalmente, es la función cuadrática:

$$J_n = e_n^2$$

Existe un procedimiento iterativo para realizar esto. La superficie de error es un hiperparaboloide con un único mínimo que, lógicamente, es nuestro objetivo. Un posible procedimiento sería situarnos sobre un punto de esa superficie y movernos en la dirección del mínimo, es decir:

$$F_{n+1} = F_n + D_n$$

donde F_k es el vector de coeficientes del filtro en el instante k , $F_k = [F_k(1), \dots, F_k(L)]$ y D_k es la dirección

donde se encuentra el mínimo. Aplicando cálculo vectorial, esta dirección se corresponde con el gradiente de la superficie cambiado de signo, es decir:

$$D_n = -\alpha \cdot \nabla_n J$$

siendo α un parámetro conocido como constante de adaptación [1].

De acuerdo con las últimas expresiones se llega a:

$$F_{n+1} = F_n + \alpha \cdot e_n \cdot X_n$$

siendo e_k el error cometido y X_k el vector de entrada en el instante k , $X_k = [x(n), \dots, x(n-L+1)]$.

En este artículo se plantea un nuevo algoritmo para incrementar la velocidad de convergencia de los sistemas adaptativos basados en el algoritmo LMS y sus variantes con un moderado coste computacional.

Recepción del artículo en su primera versión: marzo /99
Aprobación del artículo en su versión final: octubre /99
Responsable:
Emilio Soria Olivas
Dpto de Informática y Electrónica. Facultad de Físicas.
C/ Dr Moliner, 50
46100 Burjassot, VALENCIA. ESPAÑA
e-mail: emilio.soria@uv.es

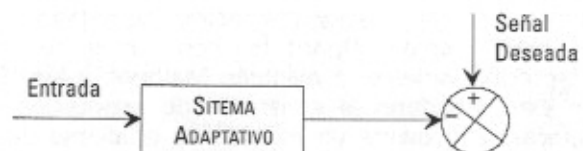


Figura 1: Esquema de un sistema adaptativo.

Este intento de incrementar la velocidad de convergencia de este algoritmo no es nuevo así, en la literatura especializada sobre el tema, nos encontramos con diferentes aproximaciones:

LMS Normalizado.

Este algoritmo modifica la constante de adaptación del algoritmo LMS básico [2]:

$$a(n) = \frac{\mu}{b + \sum_{l=0}^{L-1} x^2(n-l)}$$

con L la longitud del filtro, b constante con un valor bajo y μ un parámetro. Con b igual a 0 y μ igual a uno se garantiza la minimización del error cuadrático medio en el instante $n+1$ a partir del cometido en el instante n [3].

Momentum LMS.

Este algoritmo incrementa la velocidad del LMS y al mismo tiempo reduce las posibles oscilaciones de éste agregando a la ecuación de adaptación de pesos un término que da idea de la distancia que separa al sistema adaptativo del óptimo. La ecuación de adaptación queda [4]:

$$F_{n+1} = F_n + a \cdot e_n \cdot X_n + \beta \cdot (F_n - F_{n-1})$$

siendo β una constante (constante de momento).

El principal problema de esta variante es su mal comportamiento cerca del mínimo.

LMS con constante de adaptación variable.

Existen muchas clases de este tipo de algoritmos, en algunos, la constante de adaptación varía según el signo del gradiente. Si este signo no cambia en un número dado de iteraciones la constante de adaptación se incrementa; sin embargo, si varía, la constante se decreta [5,6]. Las razones son sencillas al analizar la superficie de error. Si el filtro adaptativo se encuentra lejos del mínimo se tendrá un conjunto de iteraciones para las que el gradiente *apuntará* en la misma dirección; por el contrario, si se encuentra cerca del mínimo es posible que a la siguiente iteración se sobrepase y, por tanto, la dirección del gradiente cambie.

Otro algoritmo de este tipo fue planteado por Kwong [7], aquí la constante de adaptación en el instante $n+1$ se determina a partir de la del instante n mediante la expresión:

$$\mu_{n+1} = a \cdot \mu_n + b \cdot e_n^2$$

siendo a y b parámetros a fijar. En este algoritmo, se fijan además unos valores máximo y mínimo para la constante de adaptación por razones de estabilidad.

Un tercer algoritmo con constante de adaptación variable lo plantean Mathews y Xie [8]. En este algoritmo la constante de adaptación se determina mediante un método de gradiente de la misma forma que se determinan los coeficientes del filtro adaptativo. La ecuación que da la constante de adaptación en el instante n viene dado por:

$$\mu_{n+1} = \mu_n + a \cdot e_{n+1} \cdot e_n \cdot X_{n+1}' \cdot X_n$$

El problema de estos algoritmos es la dificultad para determinar las acotaciones de los diferentes parámetros del sistema para que éste sea estable.

Algoritmos en bloque.

En esta clase de algoritmos la adaptación de los pesos no se realiza cada iteración sino cada cierto número de ellas (longitud del bloque) por lo que la carga computacional, número de operaciones a realizar, se reduce si la longitud del bloque es lo suficientemente grande. Al igual que ocurría en el LMS, estos algoritmos dependen de una constante de adaptación. Mikhael y Wu plantean la optimización de esta constante; posteriormente se amplía este trabajo optimizando una constante para cada peso [9].

Aparte de las variantes mencionadas, existen otras muchas [10-14] y, de hecho, periódicamente aparecen en las revistas especializadas algoritmos que incrementan la velocidad del LMS actuando sobre la constante de adaptación, proponiendo métodos computacionales eficientes para la determinación de los términos que componen el algoritmo o variantes más sencillas que incrementan la velocidad de cálculo.

En este trabajo se introduce y se analiza un enfoque diferente para resolver el problema de la velocidad de convergencia del LMS, completándose este estudio con aplicaciones típicas de sistemas adaptativos: la modelización de sistemas, la eliminación del ruido de red en un ECG y la obtención del ECG fetal.

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

Conforme los procesadores específicos de señal (DSP) han evolucionado, la velocidad de cómputo ha experimentado un espectacular aumento por lo que se originan tiempos muertos entre muestras que podrían ser aprovechados por un nuevo algoritmo para mejorar las prestaciones del sistema, lo cual resulta especialmente atractivo en el caso de los sistemas adaptativos.

Con este aumento de la velocidad de cómputo se posibilita la implementación de algoritmos con mejores características de funcionamiento pero con mayor coste computacional (p.ej. recursivos u otras variantes del LMS). Sin embargo, habitualmente se tiende a modificar lo menos posible el algoritmo ya implementado y contrastado ya que estas modificaciones suponen un coste añadido al sistema.

Estas son las razones que conducen a la solución propuesta, intercalar entre iteraciones del LMS otro algoritmo que aproveche el tiempo muerto del procesador, incrementando la velocidad de convergencia.

El algoritmo intercalado escogido ha sido el de Fletcher-Reeves [15] por dos razones fundamentalmente:

1. Es un algoritmo más rápido que el LMS ya que pertenece a la familia de los métodos de optimización de gradiente conjugado.
2. Tras unas simplificaciones, su expresión final es relativamente sencilla de analizar, estudiar e implementar.

Otros trabajos desarrollados por los autores han demostrado que la idea de encadenar algoritmos conduce siempre a incrementos en la velocidad del sistema adaptativo de partida por lo que el aprovechamiento de ese tiempo intermuestral abre un camino con enormes posibilidades para mejorar el algoritmo adaptativo [16]. En este trabajo se plantea el uso del algoritmo LMS por ser el más extendido en implementaciones *hardware*.

Antes de describir las modificaciones planteadas al algoritmo de Fletcher-Reeves se describirá brevemente el algoritmo.

Dado F_0 (coeficientes iniciales del sistema adaptativo) determínese g_0 como el gradiente de la función error para el instante 0;

- a) Tómese d_0 igual a $-g_0$
- b) Para $k=0, 1, \dots, n-1$ (n es el número de iteraciones internas)
 - c1) Hágase $F_{k+1} = F_k + \alpha_k d_k$
 - c2) Calcúlese g_{k+1} igual al gradiente de la función error en el instante $k+1$.
 - c3) Si k es diferente de $n-1$ tómese $d_{k+1} = -g_{k+1} + \beta_k d_k$ donde:

$$\beta_k = \frac{g_{k+1}' g_{k+1}}{g_k' g_k}$$

- d) Sustitúyase F_0 por F_n y vuélvase al paso a).

Como tenemos dos bucles, el correspondiente al LMS básico y el del algoritmo de Fletcher-Reeves encadenado, para distinguirlos el número de iteración del LMS se indicará como subíndice en el coeficiente del filtro mientras que el del Fletcher-Reeves aparecerá como superíndice.

La primera aproximación realizada sobre el algoritmo de Fletcher-Reeves, para reducir su carga computacional, es considerar que la constante de adaptación α_k no cambia con el tiempo. La segunda tiene que ver con la expresión de los coeficientes β_k ; estos coeficientes son iguales al cociente de las normas al cuadrado de los gradientes de la función de error en el instante $k+1$ y k . En el caso del LMS la expresión del gradiente en la iteración n es:

$$\nabla_n J = -2 \cdot e_n \cdot X_n$$

Donde e_n es el error cometido por el sistema adaptativo y X_n es el vector de entrada.

La simplificación anteriormente mencionada consiste en considerar solamente el cociente sin elevar al cuadrado. Es decir:

$$\beta_k = \frac{\|\nabla_{k+1} J\|}{\|\nabla_k J\|}$$

cumpliéndose:

$$\nabla_{k+1} J = -e_n^{k+1} \cdot X_n$$

Se llega a:

$$\beta_k = \frac{e_n^{k+1}}{e_n^k}$$

No se han incluido valores absolutos para no perder la información que sucesivos cambios de signo en el error nos pueden ofrecer. La aproximación realizada se puede justificar en cuanto que se cumplen las siguientes igualdades cerca de la convergencia:

$$e_n^{k+1} \cong e_n^k \Rightarrow \beta_k \cong \beta_k^2$$

En una implementación práctica los pasos a seguir usando el algoritmo LMS como algoritmo adaptativo principal serían:

- 1) Obtención de la salida del filtro en el instante n :

$$y(n) = F_n' \cdot X_n$$

- 2) Obtención del error:

$$e_n = d(n) - y(n)$$

- 3) Actualización de los coeficientes del filtro (algoritmo LMS):

$$F_{n+1} = F_n + \mu \cdot e_n \cdot X_n$$

los coeficientes del filtro recién calculados serán los iniciales en las iteraciones del algoritmo encadenado entre iteraciones (Fletcher-Reeves). Es decir:

$$F_n^0 = F_{n+1}$$

- 4) Determinación del gradiente cambiado de signo de la función error como dirección inicial (previamente se determinará el error del sistema con los nuevos pesos):

$$d_n^0 = e_n^0 \cdot X_n$$

- 5) Para $k=0, 1, \dots, N-1$,

- 5.1) Determinación de los coeficientes del filtro

$$F_n^{k+1} = F_n^k + \alpha \cdot d_n^k$$

- 5.2) Obtención del gradiente en el instante $k+1$:

$$g_n^{k+1} = -e_n^{k+1} \cdot X_n$$

- 5.4) Obtención de la constante β_k :

$$\beta_k = \frac{e_n^{k+1}}{e_n^k}$$

- 5.5) Determinación de la nueva dirección de búsqueda:

$$d^{k+1} = \beta_k d^k - g_{k+1}$$

- 6) Se realiza la asignación:

$$F_n^N = F_{n+1}$$

El algoritmo planteado es un *momentum-LMS* en el que la constante varía dependiendo de la evolución del error cometido por el sistema. En efecto, si se aplican de forma iterativa las ecuaciones anteriormente planteadas se llega a la siguiente expresión para la actualización de los pesos:

$$W_n^{k+1} = \beta_{k-1} \cdot W_n^k + \alpha \cdot e_n^k \cdot X_n$$

donde:

$$W_n^k = F_n^k - F_n^{k-1}$$

Si el error tiende a cero, el término de momento (factor con el incremento de pesos anterior) desaparece; se soluciona entonces el problema de convergencia cerca del mínimo.

En cuanto a la estabilidad del sistema, si se desarrolla la última expresión de actualización de los pesos aplicando la definición de β_k se llega a:

$$W_n^{k+1} = \alpha \cdot (k+1) \cdot e_k \cdot X_n$$

Vemos que el algoritmo planteado se puede entender como un LMS con una constante de adaptación creciente de forma lineal entre iteraciones. Podemos aplicar de forma aproximada la condición de estabilidad definida para un LMS [17], llegando en este caso a:

$$0 < \alpha < \frac{1}{(N+1) \cdot L \cdot E[x^2(n)]}$$

siendo L la longitud del filtro adaptativo, N el número de iteraciones internas y $E[x^2(n)]$ la energía de la señal de entrada.

RESULTADOS EXPERIMENTALES.

Las prestaciones del algoritmo propuesto se han verificado en una serie de simulaciones que dan idea de su buen comportamiento. La primera prueba realizada al algoritmo es la determinación de su velocidad de convergencia en un problema de modelización de sistemas. Para ello se implementó la estructura dada por la figura 2:

Para probar la bondad del algoritmo se representa el error al cuadrado cometido por un LMS clásico y el algoritmo planteado. En la figura 3 se representa, en decibelios, la norma al cuadrado de la diferencia entre el sistema desconocido y el sistema adaptativo. Como sistema desconocido se consideró uno con longitud 10 cuyos coeficientes se generaron de forma aleatoria sumándose un ruido normal de media cero y varianza 0.1, tomando como entrada al sistema ruido blanco de varianza 0.1 (SNR igual a 20 dB). En esta simulación se ha fijado el valor de la constante de adaptación de los algoritmos LMS en 0.01 (la misma constante que para el algoritmo Fletcher-Reeves), siendo el número de iteraciones internas de este algoritmo igual a 4. La gráfica muestra los resultados obtenidos promediando una serie de 25 experimentos. Por ello, no se da la respuesta impulsional del filtro desconocido ya que en cada uno de los experimentos se generaba éste de forma aleatoria.

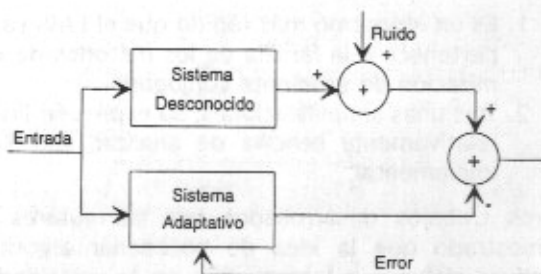


Figura 2: Esquema del sistema planteado.

Hay que destacar que en las simulaciones se fijó el número de iteraciones internas, escogiendo un valor pequeño para no incrementar excesivamente la carga computacional, y, a partir de ahí y mediante un sistema de prueba-error, se seleccionaron aquellas constantes que ofrecían un funcionamiento óptimo. Este procedimiento es el habitual a la hora de trabajar con sistemas adaptativos y redes neuronales [1].

En la figura 3 se observa que el algoritmo propuesto converge más rápido que el algoritmo LMS básico alcanzándose el mismo error en el estado estacionario.

Se podría comparar el algoritmo propuesto con otras variantes del LMS lo que daría lugar a un gran número de simulaciones. Para evitar este hecho se va a comparar con el algoritmo RLS (*Recursive Least Squares*) que presenta un coste computacional mayor que los de la familia LMS pero tiene una mayor velocidad de convergencia [1]. La simulación que se presenta a continuación es el promedio de 25 experimentos como el descrito anteriormente de modelización de sistemas. La constante de memoria para el RLS se escogió 0.99 porque era la que mejor resultado daba, tomando una constante de adaptación del LMS básico y del Fletcher-Reeves de 0.05 con 4 iteraciones internas.

En la figura 4 se aprecia que el algoritmo propuesto presenta una velocidad de convergencia parecida a la del RLS con un error en estado estacionario menor. Hay que destacar que el algoritmo RLS implica operaciones de inversión de matrices y, por tanto un alto coste computacional; en nuestro algoritmo este coste no es tan alto obteniéndose un mejor funcionamiento.

La Tabla 1 muestra la relación (en tanto por ciento) entre el número de iteraciones que el LMS y el algoritmo propuesto necesitan para converger al sistema óptimo con una precisión dada. La tabla muestra el promedio de 50 experimentos. En esta tabla se ha cambiado la constante de Fletcher-Reeves (fijando el número de iteraciones internas igual a 5) y el número de iteraciones internas (fijando la constante de Fletcher-Reeves y del LMS en 0.01).

Como se aprecia, la mejora en la velocidad de convergencia es relevante. Hay que hacer notar que este incremento no sigue una relación lineal con ninguna de las dos variables (constante del algoritmo o número de iteraciones internas).

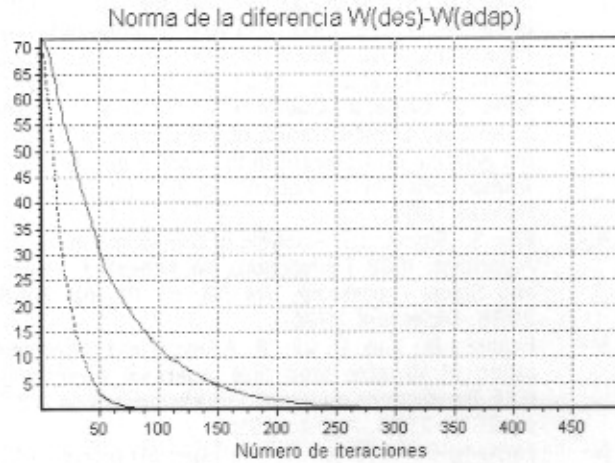


Figura 3: Resultados obtenidos con el LMS (línea continua) y el algoritmo propuesto (línea discontinua).

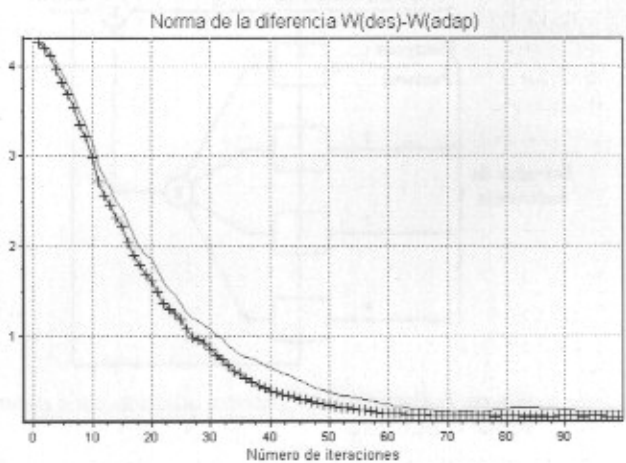


Figura 4: Resultados obtenidos con el RLS (-) y el algoritmo propuesto (+).

ITERACIONES FR	1	2	3	4	5	6	7	8	9	10
REL. ITER. (%)	67,1	34,8	26,7	20,8	15,8	9,4	7,1	6,9	4,6	3,9
CONSTANTE FR	0,005	0,0095	0,014	0,0185	0,023	0,0275	0,032	0,0365	0,041	0,0455
REL. ITER. (%)	29,4	17,7	9,3	6,2	5,3	4,1	3,6	3,9	3,2	3,3

Tabla 1. Relación de convergencia entre el LMS y el algoritmo propuesto.

La siguiente simulación realizada sobre el algoritmo fue la eliminación de ruido de red usando el esquema clásico planteado por Widrow [18], figura 5.

En la figura 6 se muestra el ECG contaminado de ruido y el resultado obtenido por el LMS clásico y el algoritmo planteado. Los parámetros de los algoritmos usados han sido: constante de adaptación del LMS y del Fletcher-Reeves iguales a 0.01 y número de iteraciones internas igual a 4.

Como se demuestra en la figura 6, el algoritmo propuesto es más rápido que el LMS clásico. Sin embargo presenta un problema en esta aplicación, la distorsión al final del complejo QRS. Esta distorsión se produce por el valor que toma el parámetro β_k en el algoritmo de Fletcher-Reeves. Este parámetro se determina mediante el cociente entre errores consecutivos de tal manera que, si el denominador es muy pequeño, el parámetro β_k tomará un alto valor alto conduciendo a la inestabilidad del algoritmo. Se puede solucionar conmutando este parámetro a cero cuando esté por encima de un determinado valor.

Por último, el algoritmo propuesto se aplicó en la determinación del ECG fetal siguiendo el esquema clásico definido por Widrow [18] (figura 7).

Para simplificar el sistema sólo se ha utilizado un electrodo de referencia. En la figura 8 se muestra los resultados obtenidos cuando se aplica un LMS y el algoritmo propuesto en la determinación del ECG fetal. Los parámetros escogidos han sido: longitud del filtro 5, constantes para el LMS y el Fletcher-Reeves iguales a 0.01 y se han tomado 4 iteraciones internas. Al igual que en las simulaciones anteriores



Figura 5: Esquema del cancelador del ruido de red.

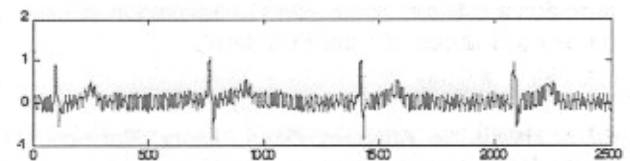


Figura 6: Aplicación del método propuesto a la eliminación del ruido de red.

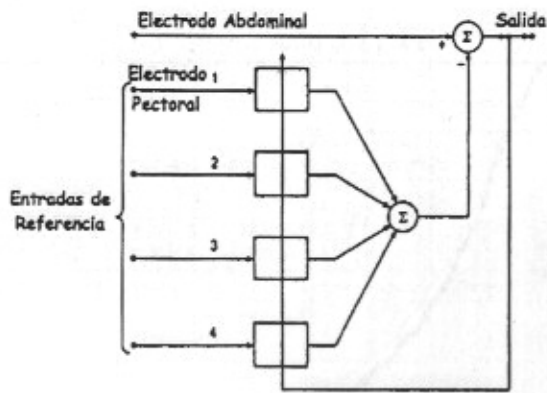


Figura 7: Esquema del sistema utilizado para determinar el ECG fetal

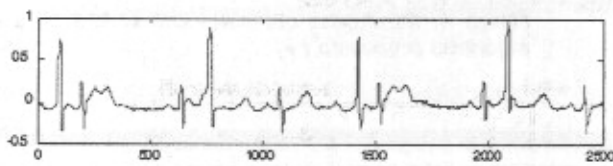


Figura 8: Aplicación del algoritmo propuesto al problema del ECG fetal

se han escogido estos parámetros mediante el procedimiento de prueba-error dentro de los límites establecidos para la estabilidad del sistema.

Como se aprecia claramente en la figura 8, el algoritmo propuesto elimina los complejos QRS maternos (los de mayor amplitud en la señal abdominal) de lo que es incapaz el algoritmo LMS clásico.

CONCLUSIONES.

En esta comunicación se ha presentado un algoritmo que incrementa la velocidad de convergencia de los sistemas adaptativos basados en el LMS usando en el aprovechamiento del tiempo muerto entre iteraciones por lo que no se requiere incrementar la potencia de cálculo del sistema. Otra ventaja de la solución propuesta es que el cambio en la programación es mínimo ya que la única operación a realizar es la inclusión del algoritmo enlazado, en nuestro caso un algoritmo de gradiente conjugado, al principal.

Este trabajo se completa con una serie de simulaciones que demuestran el buen funcionamiento del algoritmo frente al algoritmo LMS básico y el RLS que se caracteriza por su gran velocidad de convergencia. Asimismo se dan dos aplicaciones en ingeniería biomédica clásicas como son la eliminación del ruido de red y la obtención del ECG fetal.

BIBLIOGRAFÍA

1.- Haykin S. Adaptive Filter Theory. Prentice-Hall, 1995.

- 2.- Slock, D.T.M. Slock, D.T.M. On the Convergence Behavior of the LMS and the Normalized LMS Algorithms. IEEE Transactions on Signal Processing, vol 41, n° 9, pp 2811-2825, Septiembre 1993.
- 3.- Soria, E., Calpe, J., Guerrero, J., Martínez, M., Espí, J. An Easy Demonstration of the Optimum Value of the Adaptation Constant in the LMS Algorithm. IEEE Transactions On Education, vol 41, n° 1, pp 81, Febrero 1998.
- 4.- Roy, S., Shynk, J.J. Analysis of the Momentum LMS Algorithm. IEEE Transactions on Acoustics, Speech and Signal Processing, vol 38, n° 12, pp 2088-2098, Diciembre 1990.
- 5.- Evans, J.B., Xue, P., Liu, B. Analysis and Implementation of Variable Step Size Adaptive Algorithms. IEEE Transactions on Signal Processing, vol 41, n° 8, pp 2517-2534, Agosto 1993.
- 6.- Farhang-Boroujeny, B. Variable-Step-Size LMS Algorithm: New Developments and Experiments. IEE Proceedings Visual, Image and Signal Processing, vol 141, n° 5, pp 311-317, Octubre 1995.
- 7.- Kwong, R.H., Johnston, E.W. A Variable Step Size Algorithm. IEEE Transactions on Signal Processing, vol 40, n° 7, pp 1633-1642, Julio 1992.
- 8.- Mathews, V.J., Xie, Z. A Stochastic Gradient Adaptive Filter with Gradient Adaptive Step Size. IEEE Transactions on Signal Processing, vol 41, n° 6, pp 2075-2087, Junio 1993.
- 9.- Mikhael, W.F., Wu, F.H. A Fast Block FIR Adaptive Digital Filtering Algorithm with Individual Adaptation of Parameters. IEEE Transactions on Circuits and Systems, vol 36, n° 1, pp 1-9, Enero 1989.
- 10.- Pritzker, Z., Feuer, A. Variable Length Stochastic Gradient Algorithm. IEEE Transactions on Signal Processing, vol 39, n° 4, pp 997-1000, Abril 1991.
- 11.- Diniz, P., Campos, M., Antoniou, A. Analysis of LMS-Newton Adaptive Filtering Algorithms with Variable Convergence Factor. IEEE Transactions on Signal Processing, vol 43, n° 3, pp 617-627, 1995.
- 12.- Davila, C.E. Line Search Algorithms for Adaptive Filtering. IEEE Transactions on Signal Processing, vol 41, n° 7, pp 2490-2494, Julio 1993.
- 13.- Chao, J., Pérez, H., Tsujii, S. A Fast Adaptive Filter Algorithm Using Eigenvalue Reciprocals as Stepsizes. IEEE Transactions on Acoustics, Speech, and Signal Processing, vol 38, n° 8, pp 1343-1351, Agosto 1990.
- 14.- Benesty, J., Duhamel, P. A Fast Exact Least Mean Square Adaptive Algorithm. IEEE Transactions on Signal Processing, vol 40, n° 12, pp 2904-2920, Diciembre 1992.
- 15.- Jang, J.S.R., Sun, C.T., Mizutani, E. Neuro-Fuzzy & Soft Computing. Prentice-Hall, 1997.
- 16.- Soria E., Serrano, A.J., Herreros, A., Franés, J.V., Magdalena, R. Incremento de la velocidad de los sistemas adaptativos basados en el LMS implementados en DSP. Actas del 2º Congreso de Usuarios de Matlab, pp 261-266, Madrid, Septiembre 1996.
- 17.- Clarkson, P.M. Optimal and Adaptive Signal Processing. CRC Press, 1993.
- 18.- Widrow B., Glover, J.R., McCool, J.M., Kaunitz, J., Williams, C.S., Hearn, R.H., Zeidler, J.R., Dong, E., Goodlin, R.C. Adaptive Noise Cancelling: Principles and Applications. Proceedings IEE, vol 63, n° 12, pp 1692-1716, Diciembre 1975.